Discriminating between Lexico-Semantic Relations with the Specialization Tensor Model

Goran Glavaš
Data and Web Science Group
University of Mannheim
B6, 29, DE-68161 Mannheim
goran@informatik.uni-mannheim.de

Ivan Vulić
Language Technology Lab
University of Cambridge
9 West Road, Cambridge CB3 9DA
iv250@cam.ac.uk

Abstract

We present a simple and effective feed-forward neural architecture for discriminating between lexico-semantic relations (synonymy, antonymy, hypernymy, and meronymy). Our Specialization Tensor Model (STM) simultaneously produces multiple different specializations of input distributional word vectors, tailored for predicting lexico-semantic relations for word pairs. STM outperforms more complex state-of-the-art architectures on two benchmark datasets and exhibits stable performance across languages. We also show that, if coupled with a lingual distributional space, the proposed model can transfer the prediction of lexico-semantic relations to a resource-lean target language without any training data.

1 Introduction

Distributional vector spaces (i.e., word embeddings) (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) are ubiquitous in modern natural language processing (NLP). While such vector spaces capture general semantic relatedness, their well-known limitation is the inability to indicate the exact nature of the semantic relation that holds between words. Yet, the ability to recognize the exact semantic relation between words is crucial for many NLP applications: taxonomy induction (Fu et al., 2014; Ristoski et al., 2017), natural language inference (Tatu and Moldovan, 2005; Chen et al., 2017), text simplification (Glavaš and Štajner, 2015), and paraphrase generation (Madnani and Dorr, 2010), to name a few.

This is why numerous methods have been proposed that either (1) specialize distributional vectors to better reflect a particular relation (most commonly synonymy) (Faruqui et al., 2015; Kiela et al., 2015; Mrkšić et al., 2017; Vulić et al., 2017) or (2) train supervised relation classifiers using lexico-semantic relations (i.e., labeled word pairs) from external resources such as WordNet (Fellbaum, 1998) as training data (Baroni et al., 2012; Roller et al., 2014; Shwartz et al., 2016; Glavaš and Ponzetto, 2017).

Contributions. We present the Specialization Tensor Model (STM), a simple and effective feed-forward neural model for discriminating between (arguably) most prominent lexico-semantic relations – *synonymy*, *antonymy*, *hypernymy*, and *meronymy*. The STM architecture is based on the hypothesis that different specializations of input distributional vectors are needed for predicting different lexico-semantic relations. Our results show that, despite its simplicity, STM outperforms more complex models on the benchmarking CogALex-V dataset (Santus et al., 2016). Further, it exhibits stable performance across languages. Finally, we show that, when coupled with a method for inducing a multilingual distributional space (Artetxe et al., 2017; Smith et al., 2017, *inter alia*), STM can predict lexico-semantic relations also for languages with no training data available from external linguistic resources. While in this work we use STM to discriminate between four prominent lexico-semantic relations, it can, at least conceptually, be trained to predict over an arbitrary set of lexico-semantic relations, provided the availability of respective training data.

2 Related Work

Specializing distributional vectors. Given a pair of words, we cannot reliably determine the nature of the lexico-semantic association between them (if any), purely based on their distributional word vectors (Mikolov et al., 2013; Pennington et al., 2014, *inter alia*). It is a well-known property of distributional methods to conflate different types of semantic associations between words. This is why methods for specializing word embeddings for par-

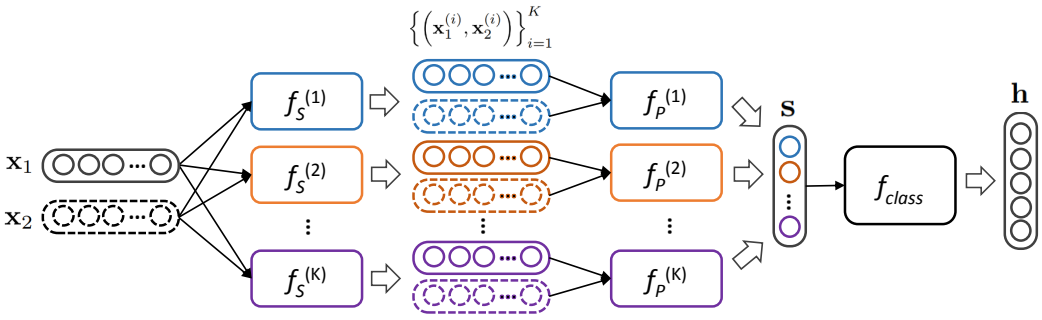

Figure 1: Architecture of the Specialization Tensor Model (STM).

ticular relations use external linguistic constraints (e.g., from WordNet) to either (1) modify the original objective of general embedding algorithms and directly train relation-specific embeddings from corpora (Yu and Dredze, 2014; Kiela et al., 2015) or (2) post-process the pre-trained distributional space by moving closer together (or further apart) words that stand in a particular relation (Wieting et al., 2015; Mrkšić et al., 2017; Vulić and Mrkšić, 2018). While these methods specialize the distributional space to better reflect properties of a particular relation, e.g., synonymy (Wieting et al., 2015; Mrkšić et al., 2017) or hypernymy (Vendrov et al., 2016; Vulić and Mrkšić, 2018), they are not able to discriminate between multiple lexico-semantic relations at the same time, i.e., the embedding space gets post-specialized for one particular relation.

Classifying lexico-semantic relations. Supervised relation classifiers learn to either identify one particular relation of interest (Baroni et al., 2012; Roller et al., 2014; Shwartz et al., 2016; Glavaš and Ponzetto, 2017) or to discriminate between multiple relations (Attia et al., 2016; Shwartz and Dagan, 2016), using labeled word pairs from external resources like WordNet. The LexNet model (Shwartz and Dagan, 2016) combines distributional vectors with recurrent encodings of syntactic paths taken from word co-occurrences in text corpora. While adding the syntactic information boosts performance, it limits the model's portability to other languages. Attia et al. (2016) train a convolutional model in a multi-task setting, coupling multi-class relation classification with binary classification of word relatedness. Unlike LexNet, this model requires only distributional vectors as input. Our specialization tensor model also requires only distributional vectors as input, but compared to the model of Attia et al. (2016), it has a simpler and more intuitive feed-forward architecture.

Glavaš and Ponzetto (2017) recently showed that asymmetric specialization of distributional vectors helps to detect asymmetric relations (hypernymy, meronymy). Following these findings, we hypothesize that detection of different relations requires different specializations of distributional vectors, so we design STM accordingly.

3 Specialization Tensor Model

The high-level architecture of the Specialization Tensor Model is depicted in Figure 1. The input to the model is a pair of unspecialized distributional word vectors $(\mathbf{x}_1, \mathbf{x}_2)$. Both input vectors are first transformed in K different ways with functions $f_S^{(1)}, \ldots, f_S^{(K)}$. Each pair of corresponding specializations $f_S^{(i)}(\mathbf{x}_1)$ and $f_S^{(i)}(\mathbf{x}_2)$ is then forwarded to the respective scoring function $f_P^{(i)}$. Finally, we feed the K scores obtained from K pairs of differently specialized distributional vectors as features to the multi-class relation classifier f_{class}.

3.1 Specialization Tensor

STM assumes that different word vector specializations emphasize different subsets of semantic properties of words that are more informative for predicting some lexico-semantic relations than others. In other words, we assume that a particular specialization function $f_S^{(i)}$ can be trained to transform the input vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ into vectors that encode properties suitable for predicting a particular relation, e.g., *hypernymy*. We set the specialization function $f_S^{(i)} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ to be a non-linear feed-forward network with a single hidden layer: it transforms the input vector $\mathbf{x} \in \mathbb{R}^m$ into a specialized vector $\mathbf{x}^{(i)} \in \mathbb{R}^n$:[1]

$$f_S^{(i)}(\mathbf{x}) = \tanh\left(\mathbf{W}_S^{(i)}\mathbf{x} + \mathbf{b}_S^{(i)}\right)$$

[1] We have also experimented with more hidden layers but $f_S^{(i)}$ with a single hidden layer yielded best performance.

with $\mathbf{W}_S^{(i)} \in \mathbb{R}^{n \times m}$ and $\mathbf{b}_S^{(i)} \in \mathbb{R}^n$ parameterizing the specialization function. Transformation matrices $\mathbf{W}_S^{(i)}$ of different specialization functions $f_S^{(i)}$ can be seen as slices of a specialization tensor $\mathbf{W}_S^{[\mathbf{1..K}]}$ (hence the model name), coupled with the specialization bias matrix $\mathbf{B}_S = \mathbf{b}_S^{[\mathbf{1..K}]}$. The number of specialization functions K (i.e., the number of slices of the specialization tensor) is the hyperparameter of the model.

3.2 Bilinear Product Scores

Following the assumption that specialization tensor slices generate relation-specific representations, we assume that an interaction between the corresponding specialized vectors $\mathbf{x}_1^{(i)} = f_S^{(i)}(\mathbf{x}_1)$ and $\mathbf{x}_2^{(i)} = f_S^{(i)}(\mathbf{x}_2)$, produced by the i-th specialization tensor slice, generates an informative score (i.e., a feature) for classifying the lexico-semantic relation for a word pair. We produce a single feature for each pair of specialized vectors $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)})$ by non-linearly squashing their bilinear product:

$$f_P^{(i)}\left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\right) = \tanh\left({\mathbf{x}_1^{(i)}}^T \mathbf{W}_P^{(i)} \mathbf{x}_2^{(i)} + b_P^{(i)}\right)$$

with the bilinear product matrices $\mathbf{W}_P^{(i)} \in \mathbb{R}^{n \times n}$ and bias terms $b_P^{(i)} \in \mathbb{R}$ being trainable model parameters. Bilinear product matrices $\mathbf{W}_P^{(i)}$ may be seen as slices of the bilinear product tensor, $\mathbf{W}_P^{[\mathbf{1..K}]}$, coupled with the bias vector $\mathbf{b}_P = [b_P^{(1)}, \ldots, b_P^{(K)}]^T$. The final K-dimensional feature vector is then simply the concatenation of bilinear product scores, that is, $\mathbf{s} = [f_P^{(1)}, \ldots, f_P^{(K)}]^T$.

3.3 Classification Objective

As the final step, we feed the feature vector $\mathbf{s}$ to the relation classifier f_{class}, a feed-forward network with a single hidden layer:

$$f_{class}(\mathbf{s}) = \tanh\left(\mathbf{W}_{cl}\mathbf{s} + \mathbf{b}_{cl}\right)$$

with parameters $\mathbf{W}_{cl} \in \mathbb{R}^{C \times K}$ and $\mathbf{b}_{cl} \in \mathbb{R}^C$, where C is the number of lexico-semantic relations between which we are discriminating. We obtain the final prediction vector $\mathbf{h}$ by applying the softmax function on the output of the relation classification component: $\mathbf{h} = softmax\left(f_{class}(\mathbf{s})\right)$.

STM is parametrized by (1) the specialization tensor and bias matrix, (2) product tensor and bias vector, and (3) classifier parameters, i.e., $\mathbf{\Omega} = \{\mathbf{W}_S^{[\mathbf{1:K}]}, \mathbf{B}_S, \mathbf{W}_P^{[\mathbf{1:K}]}, \mathbf{b}_P, \mathbf{W}_{cl}, \mathbf{b}_{cl}\}$. Assume the training set of N triples, each consisting of distributional vectors of two words and one-hot encoding of the relation that holds between these words, $\{(\mathbf{x}_{1_k}, \mathbf{x}_{2_k}, \mathbf{y}_k)\}_{k=1}^N$. We optimize STM's parameters by minimizing the regularized cross-entropy loss (i.e., negative log-likelihood):

$$J(\mathbf{\Omega}) = \lambda\|\mathbf{\Omega}\|_2 - \sum_{k=1}^{N}\sum_{j=1}^{C} y_k^j \ln(h_k^j)$$

where h_k^j is the probability that the j-th relation holds in the k-th training example (as predicted by the model), and λ is the regularization factor.

4 Evaluation

We first describe the evaluation setup (datasets, baselines, and model optimization) and then show STM's performance on a benchmarking relation classification dataset (Santus et al., 2016). Finally, we report how STM performs for different languages and in the language transfer setting.

4.1 Experimental Setup

Datasets. We use the CogALex-V dataset from the shared task on corpus-based identification of semantic relations (Santus et al., 2016). Its train and test portions contain 3,054 and 4,260 word pairs, respectively, covering four relations (synonymy: 5.4%; antonymy: 8.8%; hypernymy: 8.6%; and meronymy: 6.1%) and randomly paired words (71.1%). CogALex-V is severely skewed in favor of random word pairs and its training portion is very limited in size. Nonetheless to the best of our knowledge, it is the only publicly available dataset for multi-class classification of lexico-semantic relations on which other models have been comparatively evaluated (Attia et al., 2016; Shwartz and Dagan, 2016).

Besides the skewed class distribution and the limited size, CogALex-V also suffers from lexical repetitiveness.[2] We have thus created an additional larger and more balanced dataset by randomly sampling triples from WordNet (Fellbaum, 1998). This dataset, termed WN-LS, contains 10,000 word pairs (approximately 2,000 pairs for each of the four lexico-semantic relations and 2,000

[2] A single word can be present in up to ten pairs (although there is no lexical overlap between the train and test data).

randomly created pairs), split by 8:2 train-to-test ratio. To support the multilingual analysis, we semi-automatically translated the whole English (EN) WN-LS dataset into German (DE) and Spanish (ES).[3] We additionally translated the test portion of WN-LS to Croatian (HR), as an example of a resource-lean language.[4]

Baselines. We compare STM against two baseline models. The first baseline (CONCAT) feeds the concatenation of the distributional embeddings to a feed-forward classifier with a single hidden layer:

$$\mathbf{h}(\mathbf{x}_1, \mathbf{x}_2) = softmax\left(\tanh\left(\mathbf{W}_{cl}[\mathbf{x}_1; \mathbf{x}_2] + \mathbf{b}_{cl}\right)\right).$$

The second baseline, named BILIN-TENS is an STM reduction in which we directly forward the input vectors into the bilinear product tensor $\mathbf{W}_P^{[\mathbf{1}..\mathbf{K}]}$, without being specialized. It can be seen as STM with tensor specialization slices $\mathbf{W}_S^{(i)}$ fixed to identity matrices and biases $\mathbf{b}_S^{(i)}$ to zero vectors. Comparing STM with BILIN-TENS directly quantifies the effect the specialization tensor has on relation classification performance.

Optimization. We learn the STM's parameters using the Adam algorithm (Kingma and Ba, 2015), with initial learning rate set to 0.0001. We train in mini-batches of size $N_b = 50$ and apply dropout with the retaining probability of 0.5 to all model layers. In all experiments, we find the optimal hyperparameters (the number of specialization tensor slices K, the size of the specialized vectors n, and the regularization factor λ) via grid search within the 5-fold cross-validation on the training set.

4.2 Results and Discussion

Evaluation on CogALex-V. We show performance (F_1 score for all relations and micro-averaged F_1) on the CogALex-V dataset in Table 1.[5] For a more direct comparison with the best-performing shared task models, LexNet (Shwartz et al., 2016) and the model of Attia et al. (2016), we used 300-dimensional GloVe (Pennington et al., 2014) distributional vectors as input.

Although not by a wide margin, STM outperforms both best-performing models from the CogaLex-V shared task (Attia et al., 2016; Shwartz et al., 2016), which is encouraging, given that STM is simpler than both of these neural architectures. STM outscores the model of Attia et al. (2016), which uses the same input signal (i.e., only distributional vectors) across the board. Overall, STM also slightly outperforms LexNet (Shwartz and Dagan, 2016), despite the fact that LexNet additionally employs rich syntactic signal. STM's 6-point edge over BILIN-TENS demonstrates the effectiveness of multiple vector specializations, since this performance gain can only be credited to the specialization tensor $\mathbf{W}_S^{[\mathbf{1..K}]}$. *Antonymy* is the relation for which STM yields largest gain with respect to other models.

Model	SYN	ANT	HYP	MER	All
Attia et al. (2016)	20.4	44.8	49.1	49.7	42.3
LexNet (2016)	**29.7**	42.5	**52.6**	49.3	44.5
CONCAT	10.9	28.5	34.8	32.9	27.4
BILIN-TENS	15.7	40.3	47.9	43.3	38.9
STM	22.1	**50.4**	49.8	**50.4**	**45.3**

Table 1: Performance on the CogALex-V dataset.

Model	Lang.	SYN	ANT	HYP	MER	All
LexNet	EN	57.6	77.8	**65.9**	**83.3**	70.9
STM	EN	**58.6**	**86.6**	63.5	79.5	**72.5**
STM	DE	48.0	79.6	55.9	78.6	66.0
STM	ES	52.3	80.5	62.6	78.8	68.6

Table 2: STM performance for three languages on (respective translations of) the WN-LS dataset.

Multilingual comparison. Table 2 displays classification performance for English, German, and Spanish on respective variants of the WN-LS dataset. On the EN WN-LS version we compare STM's performance against the LexNet model (Shwartz and Dagan, 2016). To allow for a more transparent comparison of STM's performance across languages, we employed 300-dimensional English, German, and Spanish `fastText` embeddings (Bojanowski et al., 2017), pre-trained on Wikipedia.[6] STM slightly outperforms the more complex LexNet model (Shwartz and Dagan, 2016) on the WN-LS dataset as well. We believe STM's (not drastically) lower scores for German and Spanish are due to (1) distributional

[3] We first translated the dataset automatically with Google Translate and then manually fixed the translation errors.

[4] We make WN-LS dataset publicly available, together with the implementation of the specialization tensor model, at https://github.com/codogogo/stm.

[5] Optimal STM config.: $K = 5$, $n = 300$, and $\lambda = 0.001$.

[6] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

Train	Test	SYN	ANT	HYP	MER	All
EN	DE	39.1	66.8	49.3	67.6	**55.1**
EN	ES	41.7	73.0	52.6	69.0	**58.6**
EN	HR	30.5	64.7	49.1	60.5	**51.5**
DE	EN	34.0	68.6	47.2	62.4	54.2
DE	ES	39.1	61.9	44.4	60.5	50.6
DE	HR	30.3	59.8	37.7	51.7	45.2
ES	EN	47.9	74.9	46.4	68.2	**59.9**
ES	DE	37.8	66.7	47.9	62.7	53.3
ES	HR	36.1	62.2	44.6	61.5	51.4

Table 3: Zero-shot cross-lingual transfer. Best performance for each test set is shown in bold.

vectors built from smaller corpora (ES and DE Wikipedia being smaller than EN Wikipedia) and (2) language-specific phenomena (e.g., a large number of compounds in German).

Zero-shot language transfer. Finally, we investigate whether a pre-trained STM model can be leveraged to predict lexico-semantic relations for a new language, from which it has observed no training instances. We are particularly interested in such zero-shot language transfer for resource-lean languages, for which resources like WordNet do not exist. To enable transfer experiments, we needed to induce a shared bilingual (or multilingual) vector space. In all experiments, we induced the shared distributional spaces using the mapping approach and translation matrices from Smith et al. (2017).

In the first set of transfer experiments, we trained STM on the WN-LS train portion in one language (EN, ES, or DE) and evaluated it on the test WN-LS portions of all other languages, including Croatian as a resource-lean language. We show the results of these experiments in Table 3. Performance drops, compared to respective monolingual settings (i.e., performance of models trained on the WN-LS train set of the same language, see Table 2), range between 10% (EN→ES compared to monolingual ES results) and 18% (DE→EN performance compared to monolingual EN performance). These drops in zero-shot language transfer are due to imperfect bilingual embedding spaces. In fact, language transfer results seem to be very correlated with the quality of corresponding embedding translation matrices (highest for transfers between EN and ES and lowest for DE→HR transfer).[7] It is encouraging that we can build a reasonable relation classifier even for a resource-lean language, without a single training instance for that language.

[7]For example, Smith et al. (2017) report P@1 bilingual lexicon extraction performance of 73% for ES-EN, 61% for DE-EN, and 55% for HR-EN.

Train	Test	SYN	ANT	HYP	MER	All
EN+ES	HR	31.7	59.8	**52.3**	**68.3**	**54.4**
EN+DE	HR	29.0	61.7	46.5	65.3	51.5
DE+ES	HR	**36.6**	61.4	47.5	65.7	53.1
EN+ES+DE	HR	36.5	**64.6**	51.2	64.7	54.1

Table 4: Language transfer results on the HR WN-LS. Training on combinations of EN, ES, and DE data.

Finally, we examine whether we can improve prediction performance for a resource-lean language (i.e., Croatian) by combining the training data from multiple resource-rich languages (i.e., English, German, and Spanish). We show the results for this experiment in Table 4. By combining training data from different resource-rich languages, we further improve prediction performance for a resource-lean language. Compared to the EN→HR transfer, we observe 3% overall performance gain when training on merged (EN+ES and EN+ES+DE) datasets. ES and DE training instances are, however, merely translations of the original EN instances, i.e., there is no additional external knowledge being introduced. We thus believe that the observed gains are due to additional regularization provided by the multilingual training provides, which allows us to learn a model that better generalizes across languages.

5 Conclusion

We have presented a novel neural architecture for predicting lexico-semantic relations between words. The proposed tensor-based specialization model specializes distributional vectors in multiple ways and then uses these specializations to compute features for relation classification. We have demonstrated that our model outperforms more complex and resource-heavier models on two benchmarking datasets. We have further shown that our model is by design portable across languages and that it supports zero-shot knowledge transfer to resource-lean languages. As future work, we plan to experiment with more advanced neural architectures and finer-grained relations. We also intend to port the model to more languages.

Acknowledgments

Ivan Vulić is supported by the ERC Consolidator Grant LEXICAL (no. 648909).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 451–462.

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. CogALex-V shared task: GHHH-detecting semantic relations via word embeddings. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 86–91.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 23–32.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, and Diana Inkpen. 2017. Natural language inference with external knowledge. *arXiv preprint arXiv:1711.04289*.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1606–1615.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1199–1209.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 63–68.

Goran Glavaš and Simone Paolo Ponzetto. 2017. Dual tensor model for detecting asymmetric lexico-semantic relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1757–1767.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2044–2048.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Nikola Mrkšić, Ivan Vulić, Diarmuid O'Seaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics (TACL)*, 5:309–324.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Petar Ristoski, Stefano Faralli, Simone Paolo Ponzetto, and Heiko Paulheim. 2017. Large-scale taxonomy induction using entity and word embeddings. In *Proceedings of the International Conference on Web Intelligence*, pages 81–87.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1025–1036.

Enrico Santus, Anna Gladkova, Stefan Evert, and Alessandro Lenci. 2016. The CogALex-V shared task on the corpus-based identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 69–79.

Vered Shwartz and Ido Dagan. 2016. CogALex-V Shared Task: LexNET-integrated path-based and distributional method for the identification of semantic relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 80–85.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2389–2398.

Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Marta Tatu and Dan Moldovan. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 371–378.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ivan Vulić and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 56–68.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics (TACL)*, 3:345–358.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 545–550.